

**REMARKS**

In a final Office Action dated October 18, 2006, the Examiner rejected claims 1-3, 9-12, and 18-20 under 35 U.S.C. §103(a) as unpatentable over Hironaka, et al. (US 2004/0088489 A1) and Emer et al. (US 5,933,860); and rejected claims 4-8, and 13-17 under 35 U.S.C. §103(a) as unpatentable over *Hironaka* and *Emer* in view of Winberg et al. (US 2004/0199752 A1).

The claims are unamended. Applicant respectfully traverses the Examiner's rejections herein.

In response to the previous office action, applicant explained the purpose of his invention and the differences between the claimed invention and the cited art, particularly *Hironaka*. This discussion is still pertinent to the issues herein, and is incorporated herein by reference without necessarily repeating everything previously said verbatim.

As explained previously, applicant's invention is intended for use in a processor having the capability to dispatch multiple instruction for execution in parallel in each execution dispatch cycle, and in which the processor chooses instructions for dispatch from among an eligible set of instructions. Specifically, an instruction stream is sequential, but there are typically some instructions which can be executed out of order. The instruction unit identifies execution dependencies and sets of non-dependent instructions, in which the instructions within the set can be executed in any order. The instruction unit selects instructions for dispatch from among this set of non-dependent instructions.

Although certain loads and stores in a set of non-dependent instructions may be executed in any order, the hardware will typically limit the number of such loads and stores which can be

dispatched for execution in the same cycle. I.e., in order to support dispatch of two load/store operations in a cycle, the hardware must be able to support simultaneous access to two arbitrary locations in the cache. This usually requires significant additional hardware.

In accordance with the preferred embodiment, the cache is divided into multiple banks, each having its own port (or ports) for access independently of the other banks. As long as instructions access different banks, there is no conflict and they can be dispatched simultaneously. However, because most loads and stores require that the target address be computed and/or translated, the instruction unit can not know in advance which bank will be accessed. Applicant's instruction unit therefore uses a bank predict field in the instruction to guess the cache bank in which the target data resides, and select instructions for dispatch based on that guess.<sup>1</sup> I.e., it avoids dispatching two instructions which predict access to the same bank.<sup>2</sup>

Therefore, a key aspect of applicant's invention is the fact that a bank predict field in the instruction is *used by the instruction unit to select multiple load/store instructions for simultaneous dispatch* in the same execution cycle from a set of non-dependent instructions. It avoids dispatching instructions which would predict access to the same bank (and therefore constitute an address conflict). This critical feature is neither taught nor suggested by the cited art.

---

<sup>1</sup> The bank prediction is an heuristic which is not 100% accurate; in the event the bank prediction is wrong, conflict detection circuitry will detect the conflict and take appropriate action, although this is generally at the loss of one or more execution cycles.

<sup>2</sup> To be more precise, this is true where there is only one port per bank. In the more general case, where each bank could have an arbitrary number of independent access ports, it avoids dispatching more instructions to the same bank than there are independent ports.

*Hironaka* discloses a multi-bank integrated instruction and data cache, in which respective portions of the cache are allocated to instructions and the portions are variable to accommodate different operating environments. *Hironaka* discloses the capability to speculatively fetch multiple instructions simultaneously from the cache for possible execution, based on branch prediction logic.

*Hironaka* discloses a “branch predictor” which predicts a branch path of instructions, which is used by the processor to speculatively fetch the predicted branch path instruction sequence. However, as explained in response to the previous office action, *Hironaka*’s “branch predictor” does not predict anything with respect to the target of any instructions which themselves access cache, and is therefore not a “bank predict value” as recited by applicant. Furthermore, *Hironaka*’s “branch predictor” is used to select instructions for speculative fetching from the cache, and not to select instructions for execution (i.e. for dispatch to the execution unit). For these reasons, *Hironaka* fails to teach or suggest essential claim limitations.

In the latest office action, the Examiner cites *Emer* as providing the missing elements of applicant’s claims. *Emer* likewise fails to teach or suggest the recited limitations.

*Emer* discloses a type of multi-probe cache having multiple banks, each of which is direct-mapped from an address. The cache is an instruction cache, i.e., it is used for storing instructions. A portion of a memory address decodes to an index for accessing a cache line within a bank. However, being direct-mapped, there is only one cache line at each index within a bank. If the desired data is not at this location, the same index can be used to access each of the other banks, but this takes time. In order to speed up cache access, a “prediction value” is associated with certain instructions. The “prediction value” predicts the bank of cache in which next instruction

addresses will be stored. This “prediction value” increases the probability of accessing the correct I-cache bank in the first instance, thus speeding up cache accesses.

The Examiner apparently reasons that *Emer*’s “prediction value” is a “bank predict value”, as that term is used by applicant, since it predicts a bank of a cache which contains something to be accessed, and that therefore *Emer* supplies the missing elements of *Hironaka*.

The problem with this argument is that it abstracts applicant’s invention to something which predicts a cache bank, and ignores the function of the “bank predict value” as claimed by applicants, which is not taught or suggested by either cited reference. Applicant’s representative claim 1 recites:

1. A digital data processing device, comprising:
  - instruction logic which selects and decodes instructions for execution;
  - execution logic which executes instructions;
  - a first cache for temporarily storing data, said first cache comprising a plurality of banks, each bank containing at least one respective access port for accessing data in the bank; and
  - wherein at least some said instructions, when executed by said execution logic, access said first cache to perform at least one of: (a) reading data from said first cache, and (b) writing data to said first cache, and wherein a respective bank predict value is associated with each of said at least some instructions accessing said first cache, each said bank predict value *predicting a bank of said first cache to be accessed by its associated instruction*; and
  - wherein said instruction logic *selects, from among a set of multiple instructions eligible to execute by said execution logic, a subset of multiple instructions for concurrent execution by said execution logic, said instruction logic using said bank predict values of said instructions to select multiple instructions which access said first cache for inclusion in said subset*. [emphasis added]

The remaining independent claims vary in scope, but all recite the key feature that a subset of instructions is selected for concurrent execution from a potentially larger set of eligible instructions by using the bank predict value.

*Emer* fails to disclose exactly the same features that applicant mentioned earlier with respect to *Hironaka*. Specifically, although *Emer* discloses a “prediction value” which predicts a cache bank, it does not predict a cache bank ***to be accessed by its associated instruction*** (i.e. an instruction for accessing cache), as recited in the claims. Rather, it predict a cache bank at which an instruction (or group of instructions) will be found. The instruction or group of instructions at this location could be any arbitrary instructions, and to the extent that they are load/store instructions which access a cache having multiple banks, there is no disclosure or suggestion in *Emer* that the “prediction value” relates to the ***bank which is accessed*** by the instruction.

Furthermore, the claims recite that the instruction logic ***selects a subset of instruction for dispatch from among a potentially larger set of instructions eligible to execute using the bank predict value***. Like *Hironaka*, *Emer*’s prediction value is used for fetching instructions from an I-cache. *Emer*’s “prediction value” is used only to make the access to the I-cache more efficient by predicting the bank of the I-cache in which the instruction to be fetched resides, thereby avoiding the need to serially access different banks to find the required instruction. There is no teaching or suggestion in *Emer* that the disclosed “prediction value” be used for selecting a subset of instructions for dispatch to an execution unit. Indeed, as *Emer* discloses, the instructions are already selected, and the “prediction value” is used only to find them in the I-cache.

It is too easy to willy-nilly combine different elements without regard to their functional relationship, but the functional relationship is crucial to the question of whether a suggestion exists to modify or combine references in order to find the claimed invention. Applicant has consistently maintained that the fundamental feature of his invention is ***the use of a bank predict value to select the instructions to be dispatched*** to the execution unit. There are indeed many references which show using predictive values of some sort for other purposes, but neither *Hironaka* nor *Emer* discloses or suggests use of any sort of predictive value for the purpose

recited in applicant's claims. For this, and all the other reasons stated above, the claims are not obvious over *Hironaka* and *Emer*.

The remaining reference, *Winberg*, is cited to show certain features of the dependent claims, particularly relating to dynamic maintenance and historic values of the bank predict value. *Winberg* discloses a system including a "value prediction unit" which is used to predict a value for purposes of speculative execution of certain instructions. Whatever *Winberg* teaches with respect to use of historic values, it does not teach or suggest a bank predict value used for selecting accessing instructions for execution, and therefore fails to disclose the critical element missing in the combination of the other references.

In view of the foregoing, applicant submits that the claims are now in condition for allowance, and respectfully requests reconsideration and allowance of all claims. In addition, the Examiner is encouraged to contact applicant's attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,

DAVID A. LUICK



By: \_\_\_\_\_  
Roy W. Truelson  
Registration No. 34,265

Telephone: (507) 202-8725